Aerodynamic Shape Optimization Benchmarks with Error Control and Automatic Parameterization

George R. Anderson[*]
Stanford University
Stanford, CA

Marian Nemec [†]
Science and Technology Corp.
Moffett Field, CA

Michael J. Aftosmis [‡]
NASA Ames Research Center
Moffett Field, CA

Results are presented for four optimization benchmark problems posed by the AIAA Aerodynamic Design Optimization Discussion Group. The benchmarks involve drag minimization for airfoils and wings, subject to geometric and aerodynamic constraints. Our design approach involves two forms of adaptation. First, the shape parameterization is gradually and automatically enriched from a coarse initial search space. Second, adjoint solutions are used to drive adaptive mesh refinement to control discretization error. The highest-resolution parameterizations and the finest, most accurate flow meshes are each used only when nearing the optimum, thus introducing greater complexity and accuracy only when necessary to further improve the design. The first benchmark is an inviscid airfoil design problem, where we reduce the drag by a factor of 10. This example also shows how the combination of progressive parameterization and tiered discretization error control can dramatically accelerate the optimization. Next, we improve the span efficiency factor of a straight wing in inviscid flow by optimizing the twist distribution. On a viscous transonic airfoil design problem, we use an inviscid optimization approach to substantially reduce the total drag of the viscous solution. Finally, we perform automatic, multistage optimization of the Common Research Model wing, managing to hold drag roughly fixed while meeting a substantially more restrictive pitching moment constraint. This work demonstrates the ability of our shape optimization system to solve representative aerodynamic design problems, using automatic flow meshing and shape parameterization refinement.

Links: Project Slides

I. Introduction

To encourage systematic evaluation of aerodynamic optimization frameworks, a suite of benchmark optimization problems is being developed by the AIAA Aerodynamic Design Optimization Discussion Group. The purpose of these benchmarks is to exercise the capabilities of aerodynamic optimization frameworks on challenging design problems. In this work we solve the benchmark problems using an adaptive shape optimization approach comprised of two basic elements:

- **Progressive shape parameterization**: We periodically and automatically refine the search space as the shape evolves.[1]
- **Discretization error control**: We monitor and control the aerodynamic objective and constraint error throughout the optimization using error-driven mesh adaptation.[2]

Through periodic enrichment of the search space, our system is able to explore the design space more thoroughly and more robustly than under a fixed parameterization approach. Discretization error control helps ensure that accurate flow solutions are driving the optimization. Taken together, these two components aim for automatic, accurate and thorough exploration of unfamiliar design spaces. Both elements increase resolution (and thus cost) only when necessary to achieve design improvement. Throughout the work, focus

[*]Ph.D. candidate, Department of Aeronautics and Astronautics; george.anderson@stanford.edu. Member AIAA.
[†]Senior research scientist, Applied Modeling and Simulation Branch, MS 258-5; marian.nemec@nasa.gov. Member AIAA.
[‡]Aerospace engineer, Applied Modeling and Simulation Branch, MS 258-5; michael.aftosmis@nasa.gov. Assoc. Fellow AIAA.

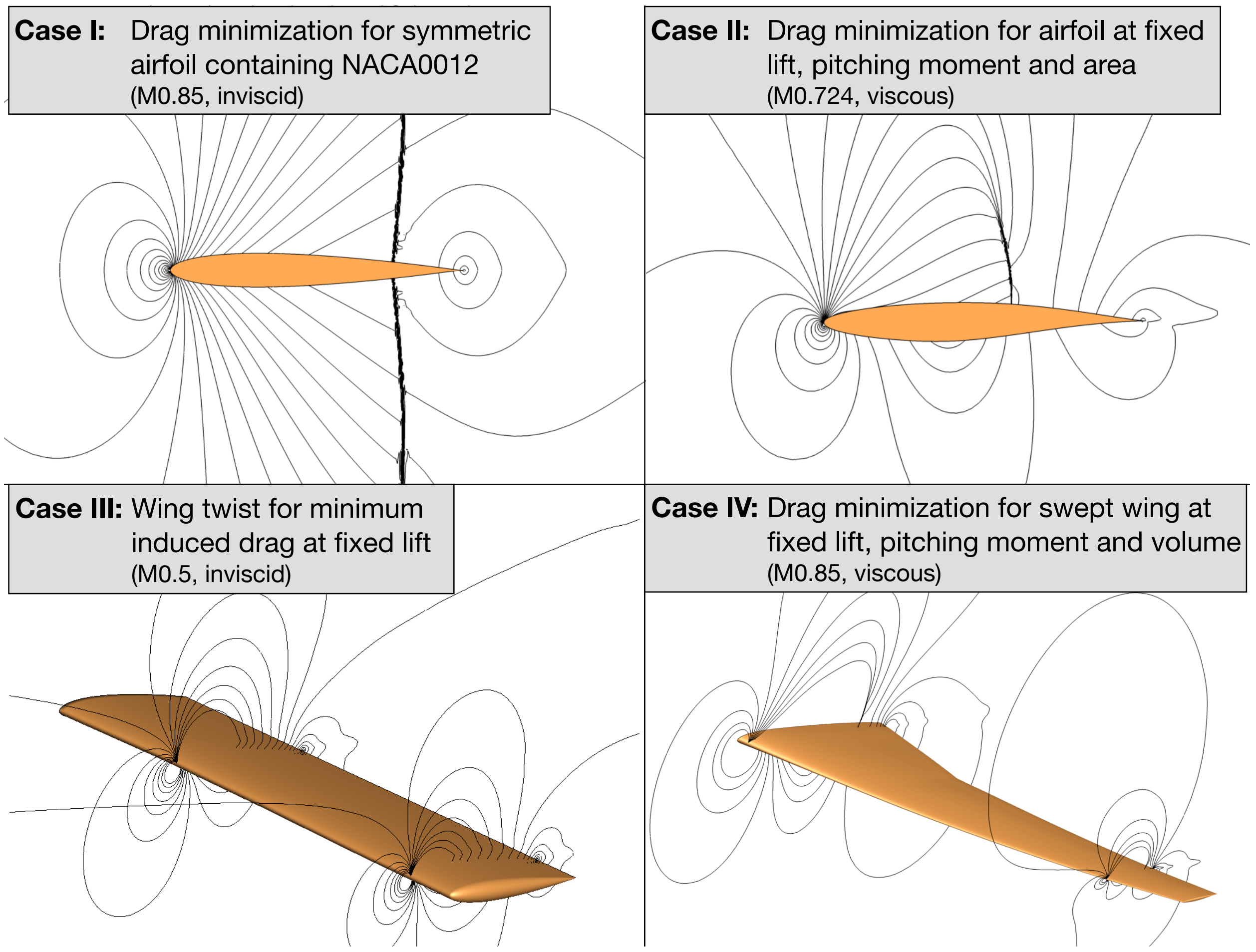

Case I: Drag minimization for symmetric airfoil containing NACA0012
(M0.85, inviscid)
Case II: Drag minimization for airfoil at fixed lift, pitching moment and area
(M0.724, viscous)
Case III: Wing twist for minimum induced drag at fixed lift
(M0.5, inviscid)
Case IV: Drag minimization for swept wing at fixed lift, pitching moment and volume
(M0.85, viscous)

Figure 1: Overview of the four benchmark cases, showing baseline designs with isomach contours.

is placed on automating non-design-related effort, such as meshing and shape parameterization, as much as possible. Additionally, our approach strives to ensure that the final optimized shape depends only on the problem specification (objective and constraints) and is robust with respect to other inputs, such as the initial design, shape parameterization, flow mesh, etc.

Figure 1 gives the essential details of the four benchmark cases, which have also been described by previous discussion group partipants.[3–8] Mach numbers range from 0.5 to 0.85, under inviscid and viscous conditions. For each case the aerodynamic optimization problem consists of finding a shape $\mathcal{S}$ that minimizes the drag $\mathcal{J} = C_D(\mathbf{Q}(\mathcal{S}))$, which is evaluated after solving the flow equations for the flow state $\mathbf{Q}$. There are also aerodynamic and geometric design constraints of the form $a \leq \mathcal{C}_i(\mathcal{S}, \mathbf{Q}(\mathcal{S})) \leq b$, which involve lift, pitching moment and wing thickness or volume.

To solve the benchmarks, we use an adjoint-based design framework[9] that uses an embedded-boundary Cartesian mesh method for inviscid flow solutions.[10,11] Adjoint solutions[12] are used for three purposes: (1) goal-oriented discretization error control via adaptive mesh refinement,[13,14] (2) aerodynamic objective and constraint gradient computation,[15] and (3) prioritization of candidate refinements of the shape control,[1] the latter being a new use of the adjoint. Design changes are driven by the SQP optimizer SNOPT.[16]

Throughout this work we optimize shapes by deforming *discrete* surface triangulations. Shape manipulation is handled with a standalone discrete geometry platform, implemented as an extension to an open-source computer graphics suite called Blender.[17] This extension allows Blender to serve as a geometry engine for optimization. For the benchmarks we use several custom deformation techniques, which are implemented as plugins to this platform. Shape sensitivities are computed analytically for each deformer. Geometric functionals (e.g. thickness and volume) are computed by a standalone tool that provides analytic derivatives to the functionals. The design framework communicates with these geometry tools via XDDM, which is an

XML protocol for extensible design markup.[9]

The discussion group places special focus on the correctness and optimality of the results. The next two sections elaborate on our two-fold approach to accurately and thoroughly explore the design space while controlling computational costs and user time. In Sections §IV and §V we present results for the four benchmark cases.

II. Progressive Search Space Refinement

The discrete surface being designed has potentially millions of degrees of freedom (one per vertex). To reduce the search to a manageable number of dimensions, the surface modifications are *parameterized*, yielding a smaller "search space", which is a subspace of the full design space. The search space is defined by a set of design variables (henceforth "DV") with values $\mathbf{X}$ and a deformation function $D(\mathbf{X})$. The local linearization of D provides the shape derivatives $\frac{\partial \mathbf{S}}{\partial \mathbf{X}}$, which describe the deformation modes of each parameter. Typically, a designer chooses a *static* set of shape deformation parameters, which may be more or less effective at improving the objective function. This search space is only a subset of the entire design space, and so it cannot generate all feasible shapes. In this paper we instead use a "progressive" parameterization approach, developed in more detail in a companion research paper.[1] Here we give a brief overview.

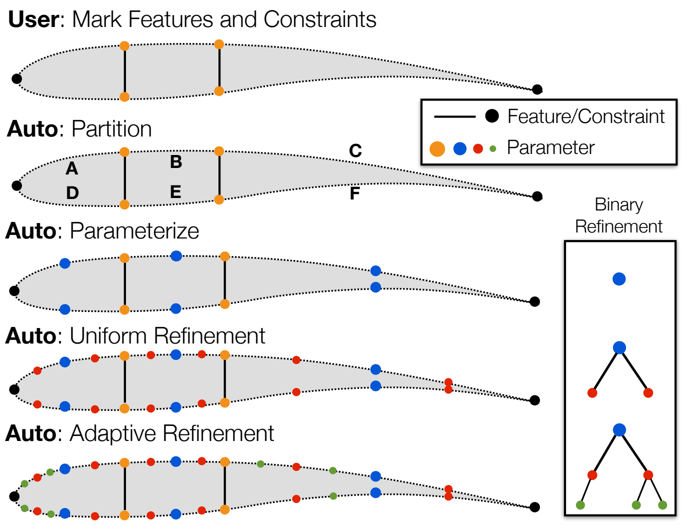


Figure 2: Progressive parameterization of an airfoil with discrete, hierarchical shape control refinement

In progressive parameterization, a *sequence* of search spaces is generated, with a progressively increasing number of design variables[a], as illustrated in Figure 2. After optimizing within an initial low-dimensional search space, the shape control is refined, opening up new avenues for improvement, and the optimization continues in the higher-dimensional space. The basic idea is to first optimize in low-dimensional search spaces, allowing rapid design improvement[b], and then to introduce more dimensions to drive towards the optimal shape.

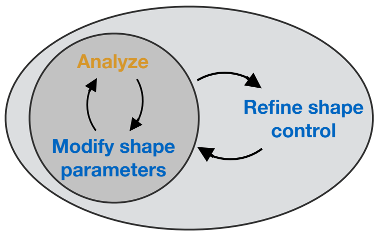


Figure 3: Optimization loop with concurrent search space refinement.

As shown in Figure 3, the optimization process is now decomposed into a series of subproblems with fixed shape control, each of which is solved by a standard aerodynamic design framework. Once the current search space is sufficiently exploited, a search space refinement request is sent to the geometry modeler. Conceptually, both the static shape optimization framework and the geometry modeler can be viewed as standalone servers, although in practice there is a fair amount of interplay among them.

A. Setup

Instead of specifying a static set of design variables, the designer marks important design *features*. For airfoil and wing design, these may involve fixing the leading edge, trailing edge or spar locations. Using these features as dividers, the surface is partitioned into regions, which are automatically parameterized. When parameterizing a curve such as an airfoil, initially a single design variable is placed at the midpoint of each region. Finer control is then gradually introduced as necessary.

[a]Another approach, which we have not yet tested, would be to redistribute existing parameters.

[b]BFGS methods theoretically converge in $\mathcal{O}(N_{DV})$ search directions, meaning that having fewer degrees of freedom generally leads to (initially) faster design improvement.

B. Parameterization Refinement

We adopt a hierarchical search space refinement technique, with a discrete approach to adding design variables, akin to h-refinement in mesh adaptation. This permits straightforward generation of a sequence of search spaces with regularly-spaced shape parameters. In the limit of refinement, this sequence converges to continuous shape control. The shape control can be encoded as a binary tree, as depicted in Figure 2. For all of the benchmark problems, we adopt the simple and robust approach of uniform parameterization refinement by binary subdivision of each tree, as shown in Figure 2.

Uniform refinement raises the issue of the rate of growth in the number of design variables, which has a critical impact on efficiency. Excessively high growth rates introduce large numbers of design variables, leading to search spaces that are slow to navigate. Although, as we will show, even uniform refinement accelerates design improvement compared to static parameterizations, uniform distribution of shape control is generally suboptimal, implying that even more gains in efficiency are possible. In the companion paper, we develop an *adaptive* refinement approach,[1] which aims to select the *most effective* parameterization, maximizing design improvement for a fixed number of design variables. This often reduces the total number of design variables required to find the optimum. Prediction of the relative effectiveness of the myriad possible shape control refinements is based on objective and constraint gradients, extracted at low cost from the final adjoint solution(s) in the previous search space. However, we show in that paper that this prediction is accurate only for well-scaled problems or when an approximation to the Hessian of the candidate shape parameters is available. For the benchmarks, we restrict ourselves to uniform refinement, briefly examining adaptive refinement only on Case I.

C. Triggering Search Space Refinement

To determine when to transition to a finer search space, we use a "trigger", or stopping criterion, that terminates the optimization in the current search space and initiates a parameter refinement. A timely and robust trigger is critical for efficiency. Over-optimization on early parameterizations leads to long periods of negligible design improvement, as also observed by other authors.[2, 18] Our approach is to only partially converge the optimization in each search space, with the goal being to move to the next parameterization when it is most computationally efficient to do so. One obvious approach is to trigger when the objective gradients have been sufficiently reduced,[c] which indicates that optimality is being approached in the current search space. However, the threshold is problem-dependent, especially with poorly-scaled search spaces, making it difficult to set. For relatively smooth problems, we adopt a more direct approach. We monitor the rate of design improvement, as measured by the slope of the objective history with respect to search directions (see, e.g. Figure 7), and trigger a parameter refinement when this slope tapers to below a certain fraction of the maximum slope achieved under the current parameterization.[d] From an engineering perspective, this makes sense. Design improvement vs. the cost to obtain that improvement is typically the figure of merit.

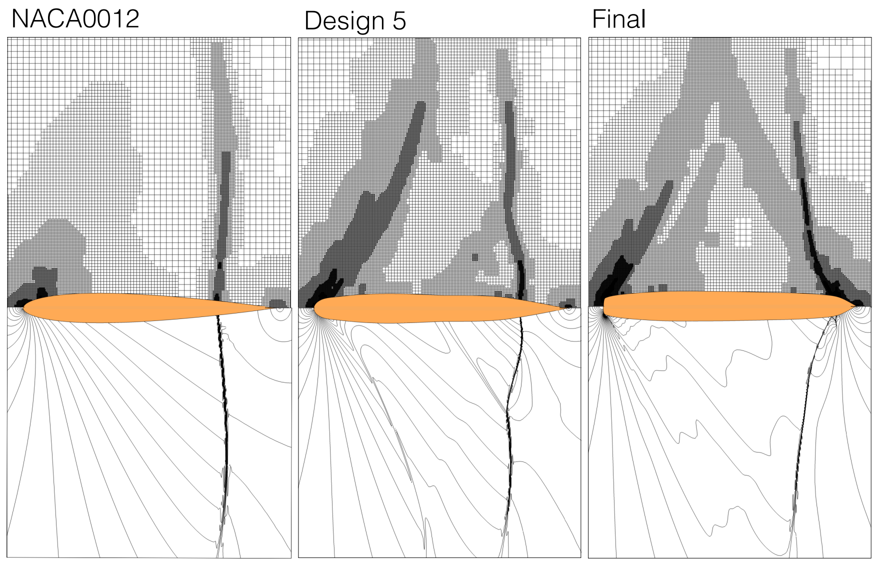
III. Discretization Error Control Strategy

Controlling discretization error is an essential component of our system, because it enables a trade between cost and accuracy that can substantially accelerate the early phases of optimization. Our approach here is somewhat atypical. As shown in Figure 4, for each design iteration, a flow mesh is automatically generated, using output-based adaptive refinement, which seeks to reduce error in the objective and constraint functionals.[2] Thus we obtain mesh convergence information for *each* design. Naturally, an indiscriminate application of tight error control would greatly increase the computational expense. However, by tracking the output error and comparing it to the evolution of the objective function, we can selectively reduce the solution accuracy during the early stages of optimization, when large design improvements are possible even with coarse simulations. The accuracy can then be automatically sharpened as the design approaches optimality. This approach also has the benefit of providing an estimate of the error on the functionals of interest throughout design.

To estimate the discretization error in an aerodynamic functional (e.g. lift or drag), we use an adjoint-weighted residual approach.[19] An example of convergence of this error estimate with mesh refinement is shown

[c]The KKT conditions are used when constraints are present.

[d]This normalization by the maximum slope handles the widely differing scales that can be present in different objective functions. For example, a drag functional is normally $\mathcal{O}\left(10^{-2}\right)$ while a functional based on operating range may be $\mathcal{O}\left(10^{5}\right)$.



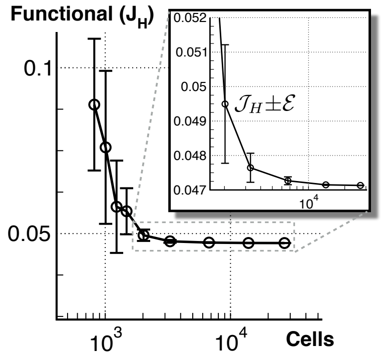


Figure 4: *Top*: Flow meshes adapted to accurately compute pressure drag of three airfoils encountered during optimization for Case I. *Bottom*: Mach contours. *Right*: Convergence of drag functional with mesh refinement for the baseline. Bars indicate uncertainty in drag and asymptotically bound the actual changes in the functional.

in the right frame of Figure 4. For most practical studies involving multiple design functionals, we construct a *combined* mesh adaptation functional that seeks to adequately resolve all the outputs. For example, in Case III (twist optimization) we adapt the mesh to resolve the span efficiency factor, which leads to a mesh that is well-balanced to compute both lift and drag.

One tremendous advantage of adaptive meshing at each design iteration is that it removes the burden of having to hand-craft flow meshes for optimization. In contrast, in the typical approach, one tries to construct a fixed mesh that anticipates how critical flow features will move as the design progresses. As the shape deviates more and more from the baseline, the fixed mesh often becomes less appropriate, leading to higher solution error as the design evolves. In our approach, accuracy is selectively *increased* while approaching the optimum. This reduces up-front costs (in both user and computational time) and also gives more credibility to the optimality of the final design.

IV. Inviscid Benchmarks

The two inviscid problems (Cases I and III) are presented first. Case I involves drag minimization for a symmetric, non-lifting airfoil. Case III is a twist optimization problem for induced drag minimization at fixed lift. Throughout each optimization we monitor objective convergence with shape control refinement, constraint satisfaction, and error in the outputs. A few ancillary details, such as optimizer settings, are given in the Appendix.

A. Case I. Symmetric Transonic Airfoil Design

The first test case involves drag minimization for a symmetric airfoil under inviscid conditions. The starting airfoil is a modified NACA 0012 (henceforth "N0012m"), where the trailing edge is made sharp.[e] The design Mach number is 0.85, while the angle of attack is fixed at $\alpha = 0°$. Additionally, the final airfoil shape must contain the original airfoil. This constraint is satisfied when $y \geq y_{N0012m}$ everywhere on the upper surface, and inversely on the lower surface. Because the solution must be symmetric, we solve the flow only in the upper half of the domain with a symmetry boundary condition at $y = 0$. The farfield boundaries are placed 96 chords away in each coordinate direction.

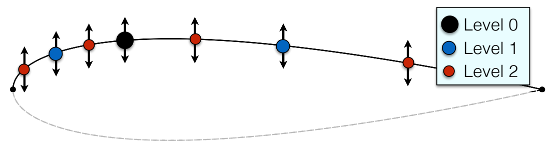


Figure 5: Case I: Initial parameterization with 7 design variables, generated by twice uniformly refining a 1-DV parameterization (lower half generated by symmetry)

[e]Via modification of the x^4 coefficient: $y = \pm 0.6\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.10\underline{\mathbf{36}}x^4\right)$

1. *Shape Parameterization*

To deform the airfoil we use a "direct manipulation" approach, where we explicitly specify the deformation of certain "pilot points" along the airfoil, as shown in Figure 5. These points serve as the design variables, while deformation of the remainder of the curve is smoothly interpolated using radial basis functions.[20–23] Each airfoil parameter has a bump-shaped deformation mode that is mostly confined to the region between its neighboring points, while maintaining smoothness. We choose the basis function $\phi = r^3$ here, primarily because it requires no local tuning parameters, making it more amenable to automation.

For this problem, initially a single pilot point is placed on the top surface, as shown in Figure 5 (black dot). Practically speaking, we observe that it is more efficient to start with several design variables, rather than a truly minimal set, so we immediately perform two uniform refinements, giving seven initial design variables. The shape control is clustered towards the leading edge by transforming the arc-length parametric space.[f] During shape control refinement, new pilot points are placed at the midpoints between existing ones. The midpoint is also measured in the transformed space, so in physical space, new parameters are biased towards the leading edge.

To handle the containment constraint, we set the lower bound of each shape parameter to the corresponding local thickness of the N0012m. The direct manipulation approach guarantees that the airfoil will exactly interpolate these pilot points. Regions between the shape control parameters may temporarily violate the containment constraint, but these violations get squeezed out as more parameters are added. In keeping with our adaptive approach, the containment constraint becomes more precise as the search space is refined.

2. *Optimization Results*

Figure 6 shows the final optimized airfoil and its pressure profile. Two features are most noticeable. First, the leading edge has become extremely blunt. In fact, after every refinement, the nose became blunter, limited only by the first shape parameter's proximity to the leading edge. This is the expected optimal result for this problem, though naturally this shape would have poor off-design performance and poor viscous performance. By the final design, the containment constraint is satisfied everywhere (not just at the interpolation points).

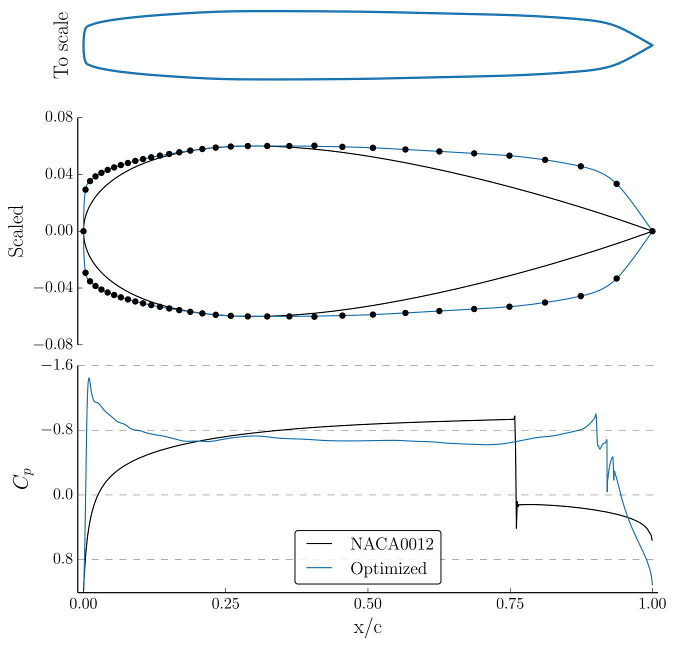


Figure 6: Case I: Final airfoil shape and pressure profile. Black dots indicate the final locations of the 31 shape control parameters.

Figure 7 shows the convergence of the objective function over 60 search directions, and over 3 parameterization levels. The parameterization was automatically refined (i.e. with no user intervention) when the objective slope tapered to 20% of its maximum slope. After each transition, a new optimization is started; no transfer of Hessian information from the previous design space is attempted. The final parameterization has 31 design variables. The drag was reduced by a factor of 10, from the baseline 471 counts down to 41.3 counts. An additional refinement to 63-DVs proved unable to further improve the design. The final design is probably close to optimal, as demonstrated by the diminishing return on each additional parameter refinement visible in Figure 7. Some further improvement is likely possible, but even the small amount of remaining discretization error combined with the very high-dimensional design space makes further improvement extremely difficult.

[f]Transformation function is $s^* = s - 0.15 sin(2\pi s)$

Table 1: Case I drag reduction with optimization. All drag measured in counts ($C_D \cdot 10^4$)

	Baseline	7-DV	15-DV	31-DV
C_D	471.3	273.8	133.0	41.3
Error estimate	± 0.1	± 0.1	± 0.1	± 0.35
Cells	26 K	49 K	50 K	61 K

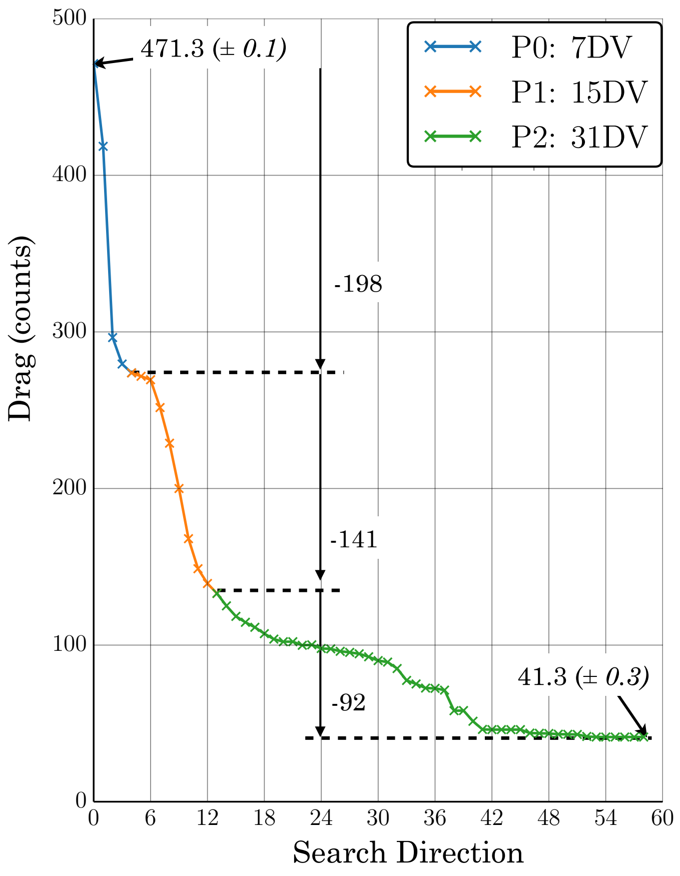


Figure 7: Case I: Objective convergence

Figure 8 compares the initial and final meshes, which were automatically adapted to reduce error in drag. Intermediate designs generated radically different mesh refinement patterns (see Figure 4 for the final design of the 7-DV parameterization level). The refinement patterns reflect movement of the shock and changes in the width of the supersonic region. For the final design, the adjoint-based mesh adaptation process provided an estimate of the remaining error in drag of about 0.3 counts ($< 3 \cdot 10^{-5}$ in C_D). The output-based mesh adaptation performed a mesh refinement study at each design iteration, yielding convergence similar to that shown in the right frame of Figure 4. This level of error was roughly constant throughout the optimization (see Table 1), giving high credibility to the final design. The cell count required to meet the error tolerance gradually increased throughout optimization. This indicates that the optimization drove the design to become more sensitive to the mesh discretization, as the shock weakened and numerical dissipation became more noticeable.

3. *Sensitivity to Farfield*

The optimization process radically increased the sensitivity of the flow to the farfield boundary distance. The initial N0012m, with its relatively confined regions of supersonic flow, is quite lenient with respect to the farfield boundary location.[g] An initial domain size study indicated that a farfield distance of 24 chords was sufficient to resolve drag to within 2 counts of the value obtained using 96-chord distances. However, the final design's carefully tuned shock structure (see Figure 6) could not be reliably resolved with farfields nearer than about 96 chords. We observed that near the final design, an inadequate farfield distance or mesh resolution can lead to an alternate solution with stronger shocks that roughly double the amount of drag! In our approach, we adapt the mesh to suit each design iteration, but always within a fixed domain size. To combat this changing sensitivity, a more comprehensive approach might periodically re-evaluate the sensitivity to farfield boundary distance, expanding the domain as necessary.

B. Assessment of the Approach

Before proceeding to the remaining benchmarks, it is worth pausing to evaluate the computational performance of each aspect of our approach. A progressive, automated approach has clear advantages in terms of *user* time and thoroughness. However, a naive implementation can also be computationally costly. In the next two sections we briefly discuss how to accelerate optimization using each adaptive component of our approach.

[g]The farfield boundary state is enforced weakly via 1-D Riemann invariants without using circulation correction.

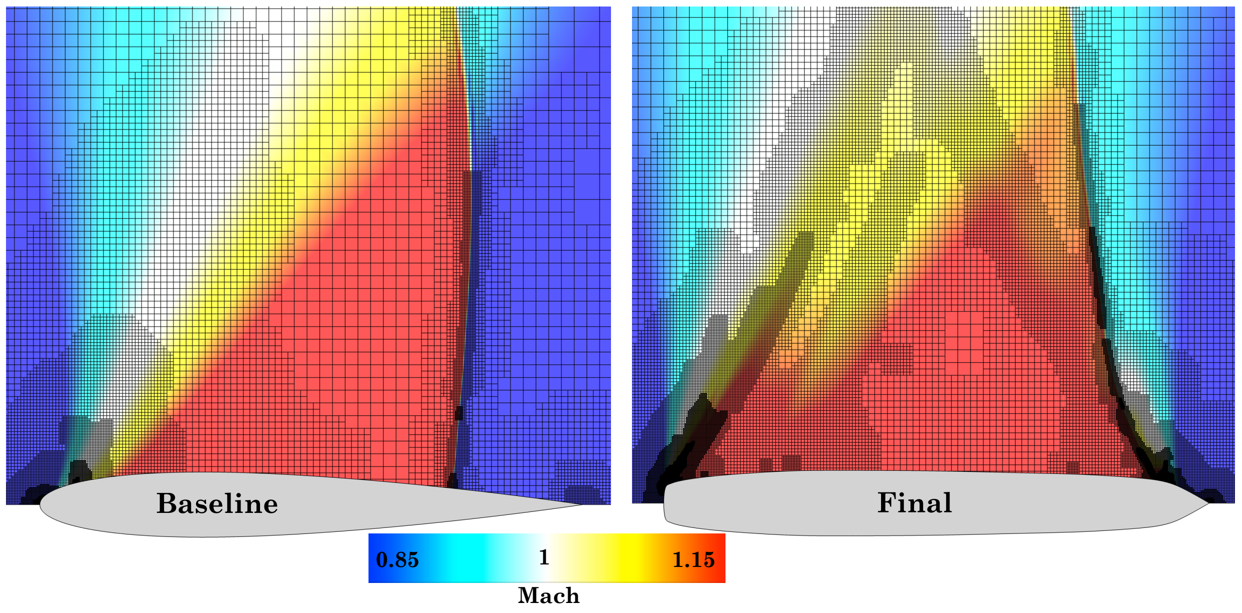


Figure 8: Case I: Comparison of baseline and final meshes. The mesh refines the regions most important for computing drag, primarily focusing on the leading edge expansion and shock. To achieve the same error tolerance for both designs, the baseline mesh required only 26K cells (upper half only), while the final design required 61K cells.

1. *Adaptive vs. Fixed Search Spaces*

Our progressive parameterization approach strongly outperforms any fixed search space on Case I. To give a rough sense of performance, Figure 9 plots design improvement versus wall-clock time for solving Case I with various parameterizations on four cores of a laptop[h]. The uniform refinement scheme (labeled "progressive") and the adaptive approach (which resulted in fewer design variables) both achieved faster and deeper overall design improvement than any coarse or fine fixed parameterization. As expected, low-dimensional search spaces support limited design improvement, while high-dimensional spaces take much longer to navigate. On the finest (63-DV) fixed parameterization, which stalled quite early, the optimizer may simply be unable to navigate the design space, as also reported by Carrier *et al.* on this problem.[4] Starting in a coarse design space appears to smooth the navigation early on, leading to a more robust search process, an observation we also expand upon in the companion paper.[1]

On this problem, the adaptive approach (which results in fewer design variables) is slightly faster than the progressive approach for most of the optimization. This speedup is largely due to the smaller number of shape derivative calls to the geometry modeler and gradient projections, and perhaps partly due to the lower dimensional design space. For slow geometry modelers, this advantage could be even

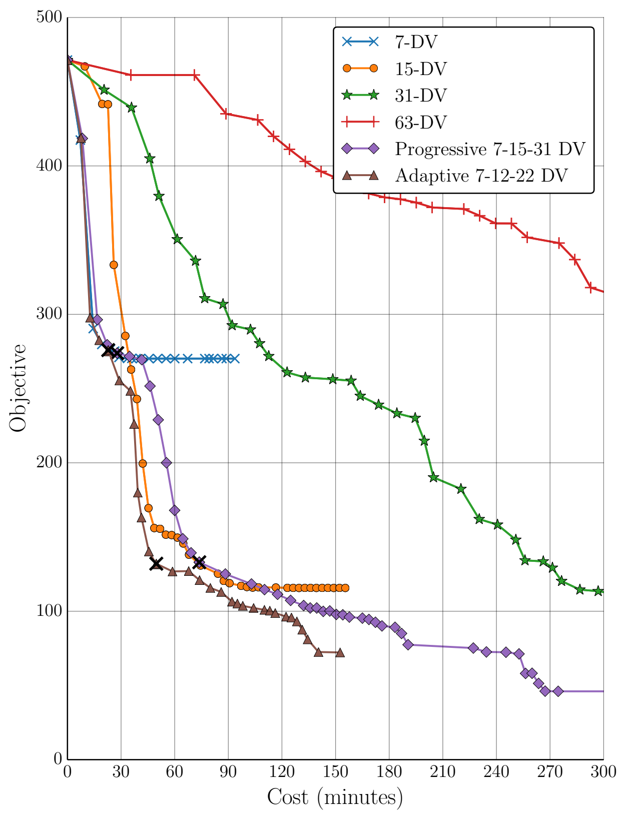
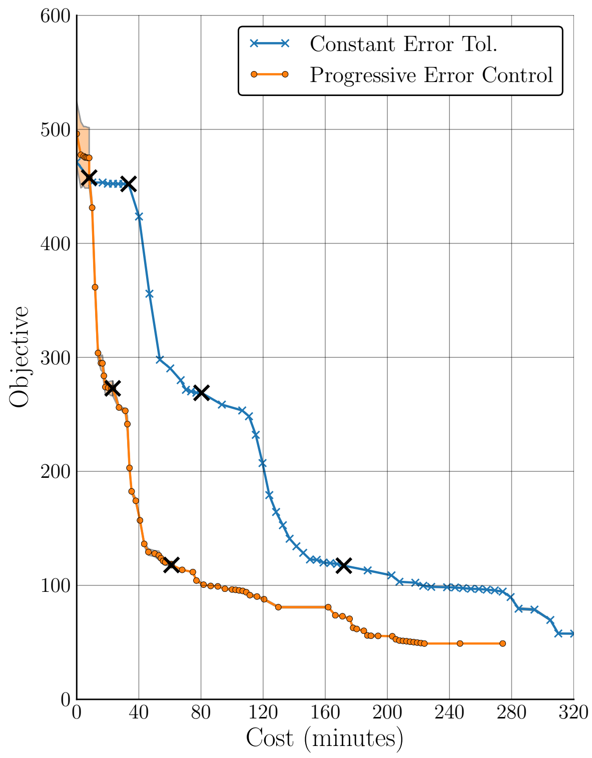
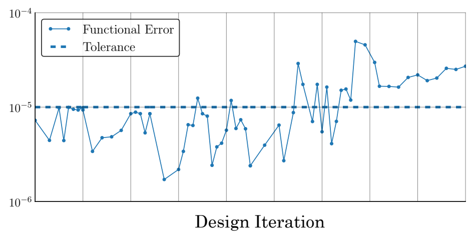


Figure 9: Case I: Cost-effectiveness of different parameterization schemes, showing design improvement vs. wall-clock time. ×-marks indicate search space refinements on the progressive and adaptive methods. All cases used identical error control settings.

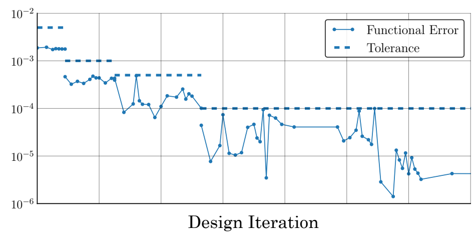
[h]2013 MacBook Pro with a 2.6GHz Intel Core i7 and 16GB of memory.



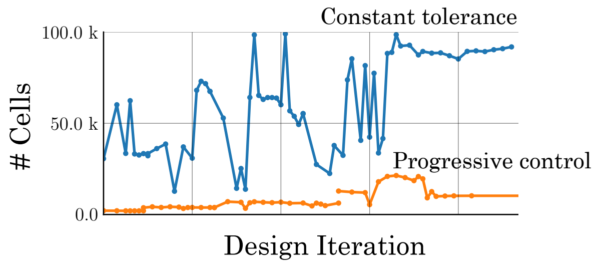
(a) Design improvement vs. wall-clock time. Only meshing, flow solver, and adjoint solutions timed: geometry generation excluded. Shaded bands on the progressive case indicate discretization uncertainty in the functional. ×-marks denote search space refinements.



(b) Constant error tolerance and actual error estimate history



(c) Progressive error control and actual error estimate history



(d) Cell count history

Figure 10: Case I: Comparison of fixed error control vs. progressive error control. Both cases were performed with identical parameterization strategies and on identical hardware (2013 MacBook Pro with a 2.6GHz Intel Core i7 and 16GB of memory).

more significant. However, factors such as the trigger, rate of variable introduction, indicator, scaling, and path-dependence make it difficult to draw firm conclusions about the potential computational advantage of adaptive refinement vs. uniform refinement from such a cursory study.

2. *Error Control Strategy*

To solve Case I, we used a constant error target throughout the optimization to satisfy the benchmark discussion group requirements of having an accurate flow solutions throughout design. Now, however, we show that the bulk of the design improvement can be obtained using quite coarse meshes, with substantial error control only being applied near the end to resolve the design landscape near the optimum. The adjoint-based mesh refinement technique used here provides a mesh refinement study and discretization error estimate along with every functional evaluation. While using tight error control throughout the optimization can lend credence to the process, blind application can result in unnecessary expense. Consulting Figure 10a, it is evident that a progressive error-targeting scheme has a significant cost advantage over the fixed-tolerance approach that we used for the Case I benchmark. Early in design, large improvements can be guided even with fairly coarse meshes. By adopting very loose tolerances early on (Figure 10c), the early stages of optimization are greatly accelerated, without sacrificing accuracy near the optimum.

Our automatic adaptive meshing approach is especially advantageous for problems like Case I, which exhibit substantial, unpredictable differences between the initial and final designs. However, near the optimum successive design iterations are often quite similar. Warm-starting the meshing and flow solutions for nearby designs is an obvious avenue for further acceleration.

C. Case III. Subsonic Wing Twist Optimization

We defer Case II momentarily to first consider the other inviscid problem, Case III. This is a wing twist optimization problem, where the airfoil section and planform remain unmodified. The objective is to minimize (induced) drag at fixed lift ($C_L = 0.375$) at a flight Mach number of 0.5.

1. Shape Parameterization

The baseline geometry is a straight, unswept, untwisted wing, generated by extruding the N0012m section three chord lengths and capping the tip by a simple revolution. For this problem we use a deformer that interpolates twist between arbitrary spanwise stations. The twist is in the streamwise plane about the trailing edge and is linearly interpolated between successive stations. Control stations can be arbitrarily spaced along the span, but for this problem we maintain strict regularity by refining only at the midpoints between consecutive stations. We allow the global angle of attack to vary and therefore hold the twist fixed at the wing root. The first parameterization ("P0") has two twist stations, located at the tip and mid-span. To generate the second level ("P1"), new twist stations are added at the midpoints between existing ones.

2. Mesh and Error Control

The baseline design has about 76.7 counts of drag. Unlike Case I, where the objective was reduced by a factor of ten, here the possible improvements are very small, which places high demands on the accuracy of the flow solution.[24] Assuming the span efficiency factor e cannot exceed 1.0, as non-planar deformations are minimal with the twist applied about the trailing edge, the minimum possible drag is roughly

$$C_{D_{min}} = \frac{C_L^2}{\pi e_0 \mathit{AR}} = \frac{0.375^2}{6.0\pi} = 74.6 \text{ counts} \tag{1}$$

However, as the wing is untapered, and twist is about the trailing edge, we do not expect that the optimal design will recover a precisely elliptical lift distribution. Additionally, we observed a very small shock on the wing tip near the trailing edge, where the flow accelerates around the tip to the top surface, which may further erode the possible drag reduction.

We compute adjoint solutions for the drag and lift functionals to compute their gradients, allowing the nonlinear lift constraint to be treated exactly by SNOPT. The error control scheduling was set to coincide with the parameterization refinements, and the farfield boundaries were placed at 48 chords away. In the first search space, the resulting adapted meshes contained about 5 million cells, while for the second search space, the meshes contained roughly 10-15 million cells to meet the tighter error tolerance.

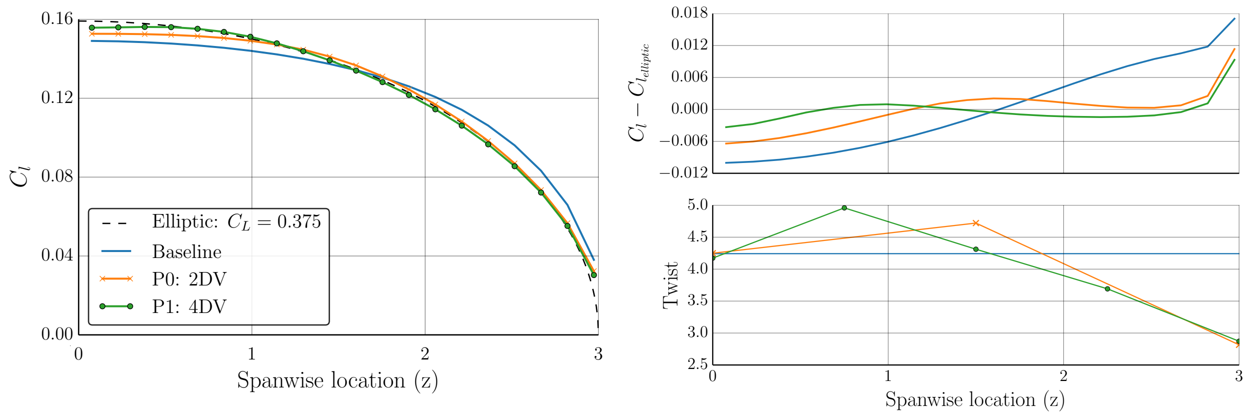


Figure 11: Case III: Twist optimization results. *Left*: Sectional lift distribution profiles. *Top right*: Deviation from elliptic distribution. *Bottom right*: Twist distribution

Table 2: Case III results.

Chords from Root	0.0	0.6	1.2	1.8	2.4	2.7	2.85	2.97	3.0
Twist (^{o})	4.2	4.8	4.5	4.1	3.5	3.2	3.0	2.9	2.9
Sectional Lift ($2c_l/b$)	0.156	0.156	0.146	0.126	0.094	0.069	0.050	0.030	0.0

3. Optimization Results

Figure 11 shows the main results of the optimization. The lift distribution rapidly approaches an elliptical shape, with only very small discrepancies at the tip, due to the untapered section, and at the root, which compensates to exactly match lift. Figure 12 shows the convergence of the lift and drag functionals. Because a coarser mesh was used in the initial design space, there is a jump in functional values when transitioning to the finer design space. By the end, lift is satisfied and drag is reduced. To accurately determine the total improvement, we performed an additional accurate analysis on the initial and final designs. Figure 13 shows the convergence of span efficiency factor (e) with mesh refinement for the initial and final designs, trimmed to $C_L = 0.3750$. The initial design had $C_D = 76.7$ counts of drag ($e = 0.973 \pm 0.005$). By the final design this was improved to $C_D = 75.6$ counts ($e = 0.987 \pm 0.003$).

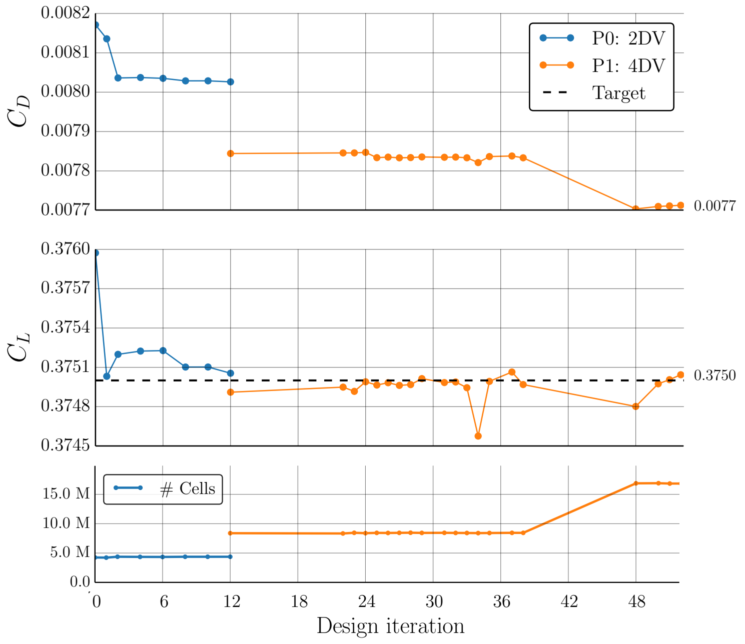
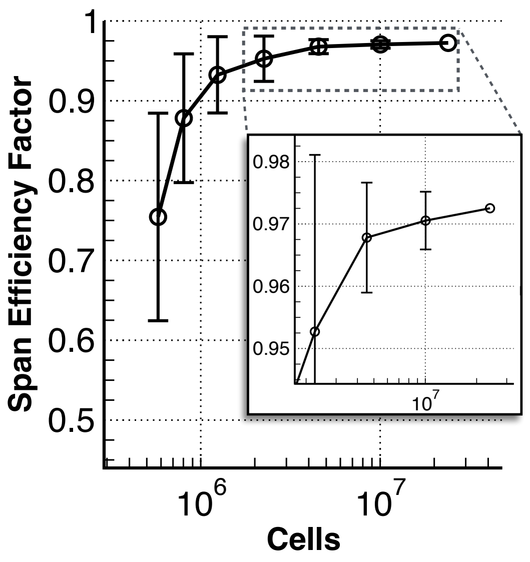
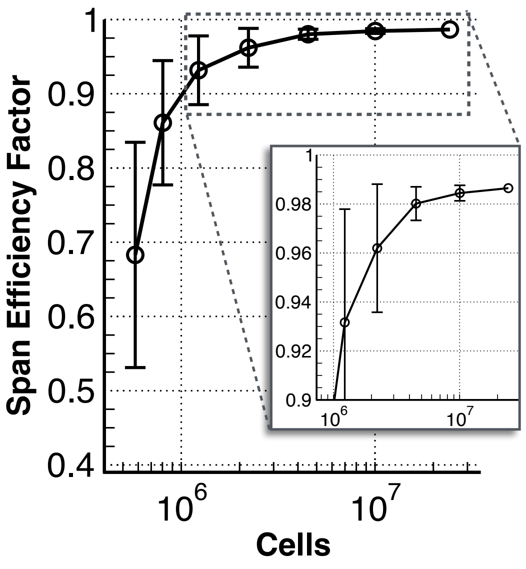


Figure 12: Case III: *Top*: Convergence of drag. *Middle*: Convergence of lift. *Bottom*: Cell count history



(a) Baseline design



(b) Final design

Figure 13: Case III: Convergence of span efficiency factor with mesh refinement

V. Viscous Benchmarks

We now turn to the two RANS optimization benchmarks. As our design framework uses an inviscid solver, the results will not be directly comparable to other viscous results. For Case II, we modify the design problem slightly to achieve better viscous performance with an inviscid optimization approach. The results for Case II are verified under viscous conditions using a recently developed 2D Cartesian RANS approach by Berger and Aftosmis.[25]

A. Case II. Transonic Airfoil Design

Case II revisits transonic airfoil design (Mach 0.734), but this time with more realistic design constraints. The objective is again to reduce the drag, while constraints are imposed on lift, pitching moment (which is initially violated) and the area A:

$$C_L = 0.824$$
$$C_M \geq -0.092$$
$$A \geq A_{RAE} \approx 0.07787c^2$$

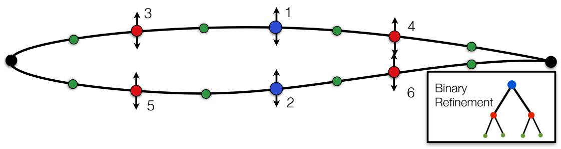
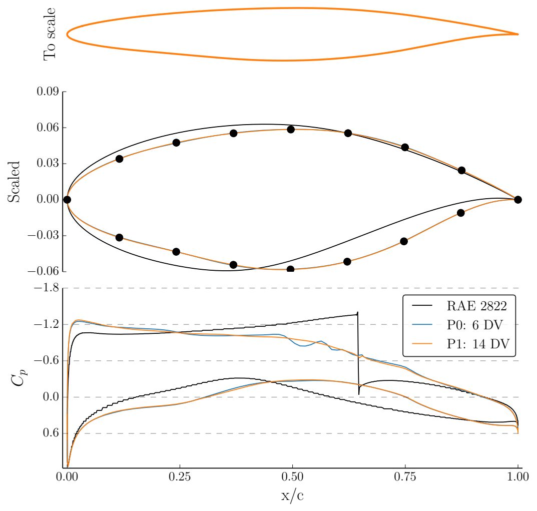
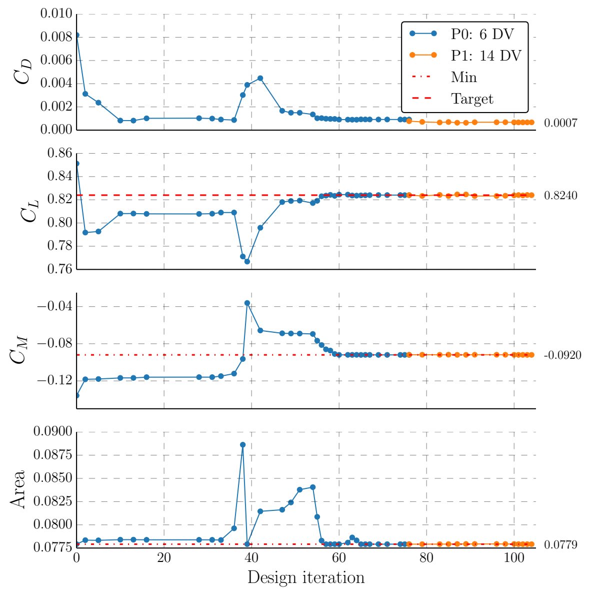


Figure 14: Case II: Initial 6-DV parameterization and uniform refinement.

The baseline shape is the RAE 2822 airfoil. We parameterize the deformation with the same curve deformer as in Case I. In addition to angle of attack, there are initially six shape parameters, as shown in Figure 14 (blue and red dots). A second 14-DV design space is generated by uniform refinement (green dots in Figure 14). For discretization error control, we set a lower tolerance in the first search space, and then tighten it to ± 0.5 counts of drag on the second level. We compute adjoint solutions for the drag, lift and pitching moment functionals to compute their gradients. The area is computed on the discrete surface, and the constraint gradients are differentiated analytically.



(a) *Top*: Final airfoil to scale. *Middle*: Final parameterization. *Bottom*: Pressure profile at the end of each optimization level.



(b) Convergence of aerodynamic functionals (plotted only at major iterations). The first level used somewhat coarser flow meshes, visible in the subtle discontinuity in drag where the parameterization is refined.

Figure 15: Case II: Trial 1 (inviscid) results across two parameterization levels

1. *Optimization Trial 1 (Straightforward Inviscid Design)*

Figure 15 shows the results of driving the optimization with inviscid flow solutions at the specified flight conditions. SNOPT rapidly drove down the drag. After several search directions without noticeably improving the aerodynamic constraints, it increased its internal constraint weights, rapidly driving the pitching moment and lift to be satisfied. The shock is nearly eliminated even under the first parameterization. After refining to 14-DVs (and simultaneously tightening the discretization error tolerance), the shock is completely eliminated. An additional refinement to 30-DV's did not yield any further improvent for reducing the negligible remaining inviscid drag.

2. *Viscous Analysis*

To check the viscous performance of this design, we computed the flow using the Cartesian RANS solver mentioned above,[25] with a Spalart-Allmaras turbulence model at $Re_c = 6.5$ million. The RANS solution is shown in Figure 16. The inviscidly-designed airfoil does have superior viscous performance to the original RAE 2822. Consulting Table 3, the total drag has been reduced by 34 counts. However, the presence of the boundary layer increased the angle of attack necessary to achieve $C_L = 0.824$, resulting in higher Mach numbers over the top surface and thus the presence of a moderately strong shock.

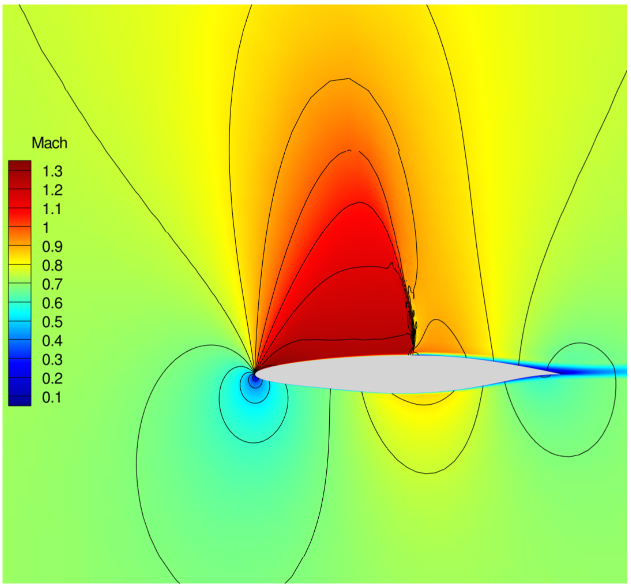


Figure 16: *Case II, Trial 1:* RANS analysis of the straightforward inviscid design, showing pressure contours and colored by Mach number. ($C_L = 0.824$, $M0.734$, $Re = 6.5$ million)

3. *Optimization Trial 2 (Trailing Edge Deflection)*

At these flight conditions under the assumption of inviscid flow, a wide range of shapes eliminate the shock while satisfying the constraints. However, most of these designs have poor viscous performance. To encourage the optimizer to prefer shapes with better viscous performance, we follow the approach used by Smith *et al.*[26] and earlier by Campbell.[27] Briefly, we mimic the shallower *effective* trailing edge slope present in the true viscous flow, by applying a small upward cubic deflection to the last 20% of the airfoil at every design iteration during inviscid design:

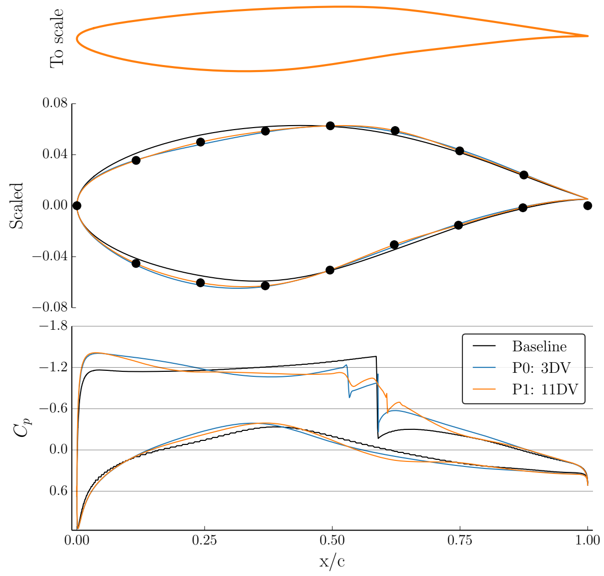
$$y = y + \left(\frac{x - 0.8}{0.2}\right)^3 sin(\theta) \tag{2}$$

where we used $\theta = 0.3°$. This forces the optimizer and inviscid solver to compensate for a shallower effective trailing edge camber line. Naturally, the fictitious deflection is then removed when analyzing the final design under viscous conditions. To help exclude irrelevant designs with poor viscous performance, we also roughly constrained the thickness at three locations by removing three thickness design variables from the initial search space.

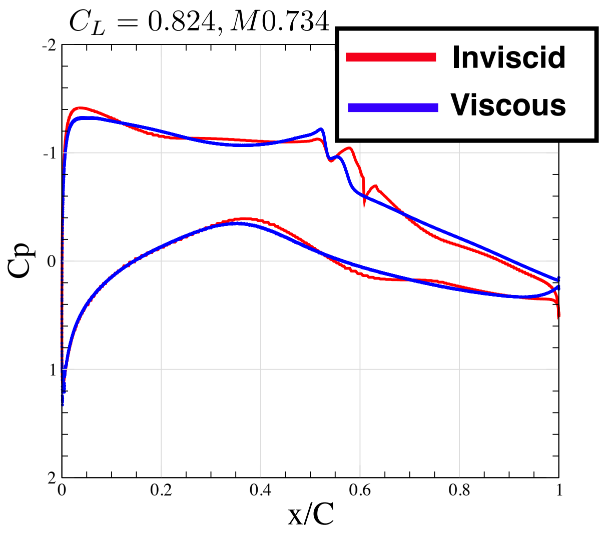
Figure 17 shows the results of this second optimization. Although the new inviscid design (top left frame) is not fully shock-free, the viscous performance (other frames) is substantially better, leading to about 124 counts of drag, or an additional 38-count reduction beyond the straightforward inviscid design in Trial 1. As show in Table 3, this new design has a much closer match between the trimmed α for the inviscid analysis and for the viscous analysis, leading to similar behavior over the front region, and importantly, similar shock placement.

Table 3: Case II drag reduction with optimization.

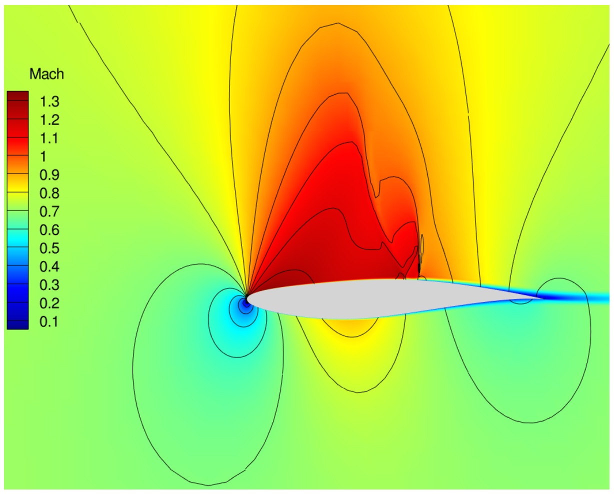
		Baseline	Trial 1	Trial 2
Inviscid	C_D	0.0068	0.0007	0.0007
	Error	±0.0001	±0.00005	±0.00006
	Cells	21K	12K	27 K
	α_{trim}	1.73°	2.28°	2.72°
Viscous	C_D	0.0196	0.0162	0.0124
	α_{trim}	2.76°	3.60°	2.61°



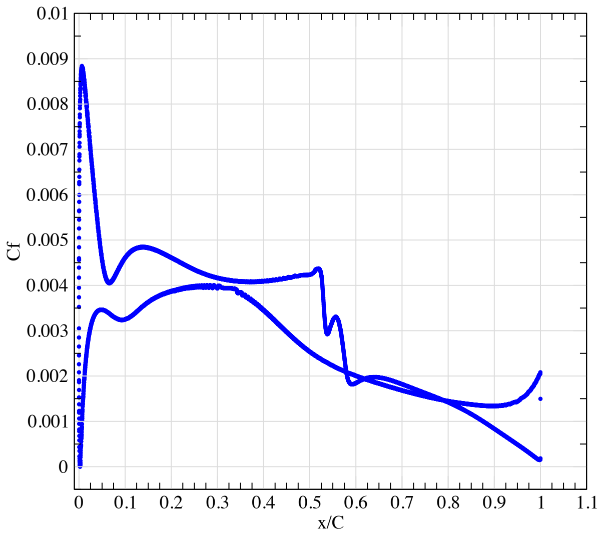
(a) *Top*: Final airfoil to scale. *Middle*: Final parameterization. (Note the deflection at the trailing edge, which is applied *after* the curve deformation.) *Bottom*: Inviscid pressure profiles



(b) Pressure profiles for final design ($\alpha_{inviscid} = 2.72°$, $\alpha_{RANS} = 2.61°$)



(c) Viscous solution of final design, showing isobars, with color indicating Mach number ($C_L = 0.824$, $M0.734$, $Re = 6.5$ million)



(d) Friction coefficient (C_f) for final design

Figure 17: *Case II, Trial 2*: Results for inviscid optimization using ficitious trailing edge deflection

B. Case IV. Transonic Wing Design

The final case is a wing design optimization problem at Mach 0.85. The objective is to reduce drag, subject to a lift constraint and a pitching moment constraint[i], which is initially violated. The baseline geometry is the Common Research Model (CRM) wing, scaled so that the mean aerodynamic chord has unit length. The planform is fixed, while variation in the vertical direction is permitted, including airfoil design and sectional twist. The twist is about the trailing edge and is fixed at the root, while the angle of attack is permitted to vary. The wing is required to maintain its initial volume V_0 and also to maintain at least 25% of its original local thickness t_0 everywhere. To approximate this continuous thickness constraint, we used a 10×10 grid of constraints distributed evenly across the planform. The full optimization problem is

$$\begin{aligned} &\min C_D \\ C_L &= 0.5 \\ C_M &\geq -0.17 \\ V &\geq V_0 \approx 0.26291 \\ t_i &\geq 0.25 t_{i_0} \forall i \end{aligned}$$

We solve this problem unmodified at inviscid conditions to demonstrate our design approach. The purpose of this example is not to directly compare the quantitative results to those achieved by groups using viscous design approaches, but rather to demonstrate our automated design system on a representative 3D problem.

1. *Shape Parameterization*

For this problem, we use a deformer similar to that used in Case III, but here it interpolates both twist and airfoil section deformation independently. At each station a curve deformer (identical to the setup used for Cases I and II) deforms the airfoil shape, after which the twist is applied. As before, the twist is in the streamwise plane about the trailing edge and is linearly interpolated. Control over airfoil sections and twist can happen at different stations, allowing for "anisotropic" shape control. For example, the twist control may have a higher spanwise resolution than the airfoil control. Similarly, each airfoil control station can offer different shape control resolution.

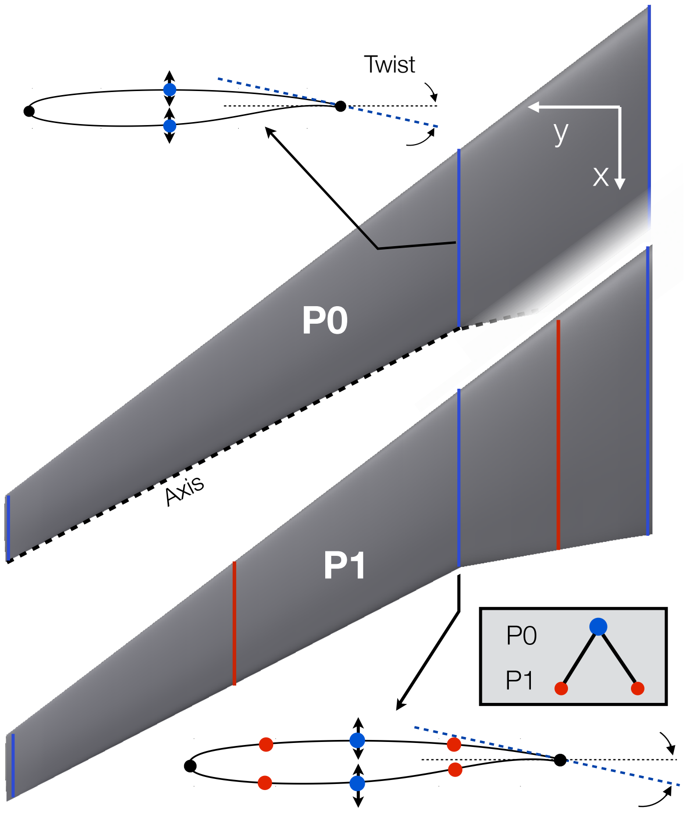


Figure 18: Case IV: First two shape control levels (8-DV and

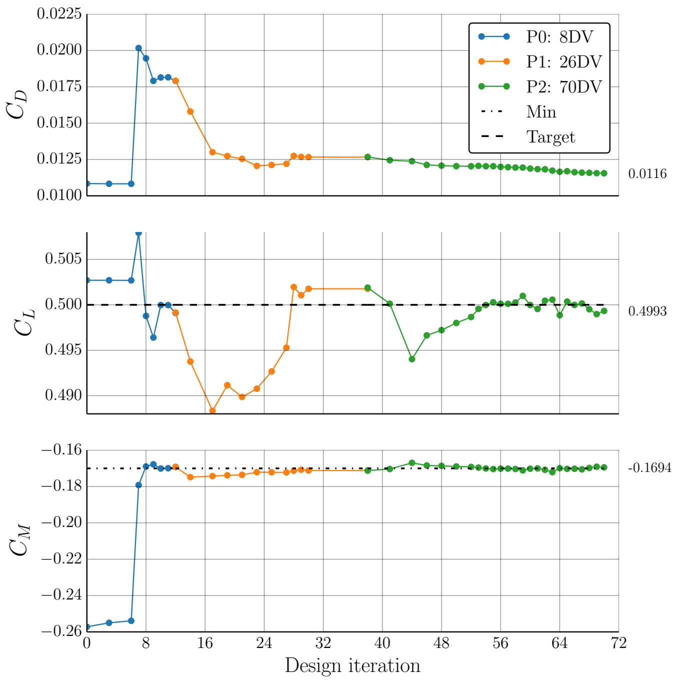
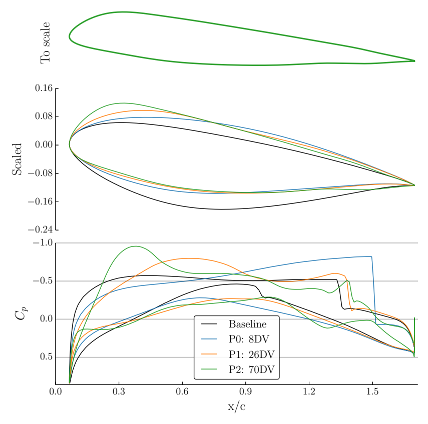
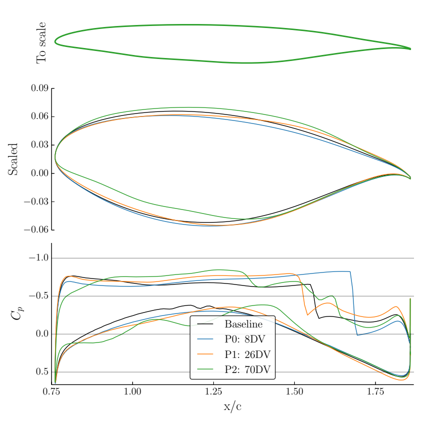


Figure 19: Case IV: Convergence of aerodynamic functionals (only plotted at successful search directions).

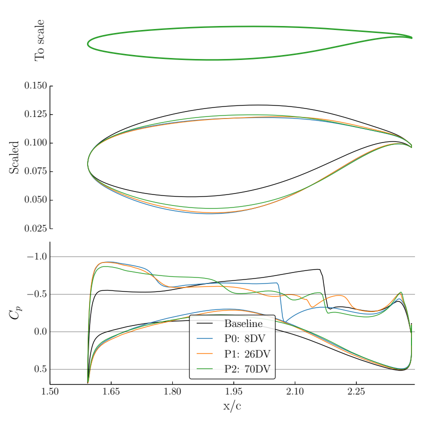
[i]Measured about the point (1.2077, 0, 0.007669) with the origin at the leading edge of the wing root.



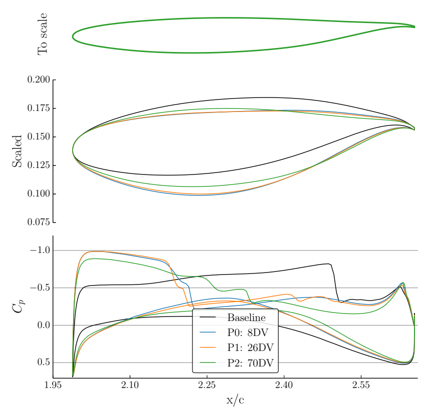
(a) 2.35%



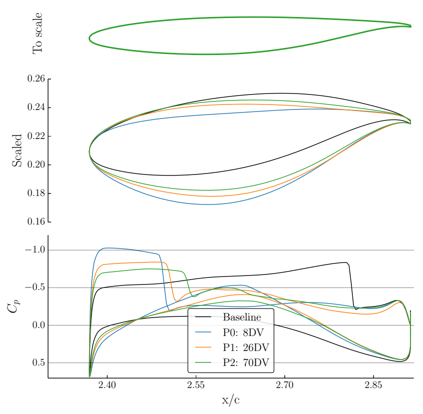
(b) 26.7%



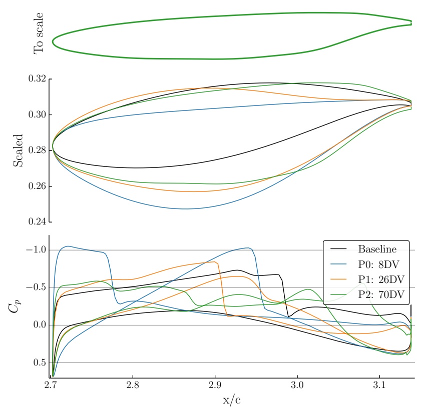
(c) 55.7%



(d) 69.5%



(e) 82.8%



(f) 94.4%

Figure 20: Case IV: Airfoil cuts and inviscid solution pressure profiles.

Figure 18 shows the first two parameterizations ("P0" and "P1"). P0 allows twist at the tip and break (fixed at the root) and rough camber and thickness control (two control points each on the root, break and tip sections). There are initially eight shape design variables, plus the angle of attack. To refine the parameterization, we add new control stations at the spanwise midpoints between the existing stations, and simultaneously uniformly refine the airfoil control at each existing station. Two additional parameterization levels ("P1" and "P2") are automatically generated when needed, with 26 and 70 geometric design variables, respectively.

2. *Optimization Results*

Figure 19 shows the convergence of the aerodynamic functionals over the three parameterization levels. Under "P0", the initially violated pitching constraint is driven to satisfaction. To do this, large airfoil deformations are enacted, as shown in Figure 20 (blue lines), with a resulting sharp increase in drag. After adding more shape control resolution, the drag is rapidly driven down almost to the initial value, while nearly satisfying the constraints. The airfoil sections (Figure 20, orange lines) relax to more subtle changes from the baseline shape. The thickness and volume constraints are met at every design iteration. All of the constraints are nearly satisfied by the end, with only a slight drag penalty associated with having to meet the initially violated pitching moment constraint.

The flow mesh was automatically adapted for every design iteration, with about 12-18 million cells, depending on the design iteration, which was adequate to drive the optimization forward. We did not yet analyze the performance of the inviscidly optimized wing with a viscous flow solver, and it is possible that a modification of the trailing edge (as in Case II) would improve the viscous performance. Nevertheless, this example demonstrates the ability of our adaptive shape optimization system to automatically solve a standard 3D aerodynamic optimization problem with constraints. After the initial problem setup, no user intervention or problem modification was required for the remainder of the optimization.

The design improved substantially at each parameterization level, including in the finest 71-DV search space. This suggests that, although the optimization may have converged with respect to the existing shape parameters, it has not yet converged with respect to the refinement of the shape control resolution. By monitoring the amount of design improvement achieved under successive search spaces, our system is able to inform a designer about convergence towards *continuous* optimality, information that is not typically available under a static-parameterization approach.

VI. Summary

We presented results for four optimization benchmark problems. On the two inviscid design problems, expected results were recovered. For Case I, our final shape is nearly identical to shapes seen by previous participants,[3–7] with similar or superior drag performance (reduction of $10\times$ from the baseline). For Case III, although the baseline shape left little room for improvement, our system optimized the wing twist, managing to drive the lift distribution closer to elliptic and reducing the drag by one count. On Case II, we showed how an inviscid design approach with slight problem modifications was able to reduce the RANS-analyzed drag by 72 counts. On Case IV, we showed how our progressive parameterization and discretization error control systems work together to automatically solve a typical 3D design problem, holding drag roughly constant while meeting an initially violated pitching moment constraint.

Our approach combines progressively refined shape spaces with progressive discretization error control. We showed how progressive parameterizations susbtantially reduced the optimization cost compared to using fine fixed parameterizations. By using a tiered approach to discretization error control, we achieved rapid early design progress on coarser meshes, while automatically transitioning to higher resolution when approaching the optimum. In the future we hope to demonstrate this system on larger scale problems, such as low-boom design or wing-body-nacelle integration.

Acknowledgments

The authors gratefully acknowledge helpful discussions with David Rodriguez and with the manuscript reviewers, Joshua Leffell and Susan Cliff. Support for George Anderson and Michael Aftosmis was provided by a 2.5-year NASA Seedling award. Support for Marian Nemec was provided both by NASA Ames Research Center contract NNA10DF26C and by the Subsonic Fixed Wing and Supersonics projects of NASA's Fundamental Aeronautics program.

References

[1]Anderson, G. R. and Aftosmis, M. J., "Adaptive Shape Control for Aerodynamic Design," *AIAA Paper 2015-0398*, Kissimmee, FL, January 2015.

[2]Nemec, M. and Aftosmis, M. J., "Output Error Estimates and Mesh Refinement in Aerodynamic Shape Optimization," *AIAA Paper 2013-0865*, Grapevine, TX, January 2013.

[3]Bisson, F., Nadarajah, S., and Shi-Dong, D., "Adjoint-Based Aerodynamic Optimization of Benchmark Problems," *AIAA Paper 2014-0412*, National Harbor, MD, January 2014.

[4]Carrier, G., Destarac, D., Dumont, A., Meheut, M., Salah El Din, I., Peter, J., Khelil, S. B., Brezillon, J., and Pestana, M., "Gradient-Based Aerodynamic Optimization with the elsA Software," *AIAA Paper 2014-0568*, National Harbor, MD, January 2014.

[5]Telidetzki, K., Osusky, L., and Zingg, D. W., "Application of Jetstream to a Suite of Aerodynamic Shape Optimization Problems," *AIAA Paper 2014-0571*, National Harbor, MD, January 2014.

[6]Poole, D. J., Allen, C. B., and Rendall, T. C. S., "Application of Control Point-Based Aerodynamic Shape Optimization to Two-Dimensional Drag Minimization," *AIAA Paper 2014-0413*, National Harbor, MD, January 2014.

[7]Amoignon, O., Navratil, J., and Hradil, J., "Study of Parameterizations in the Project CEDESA," *AIAA Paper 2014-0570*, National Harbor, MD, January 2014.

[8]Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "RANS-based Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, 2014.

[9]Nemec, M. and Aftosmis, M. J., "Parallel Adjoint Framework for Aerodynamic Shape Optimization of Component-Based Geometry," *AIAA Paper 2011-1249*, Orlando, FL, January 2011.

[10]Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," *Lectures at the Von Karman Institute*, March 1997.

[11]Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," *AIAA Paper 2000-0808*, Reno, NV, January 2000.

[12]Nemec, M., Aftosmis, M. J., Murman, S. M., and Pulliam, T. H., "Adjoint Formulation for an Embedded-Boundary Cartesian Method," *AIAA Paper 2005-0877*, Reno, NV, January 2005.

[13]Nemec, M. and Aftosmis, M. J., "Adjoint Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes," *18th AIAA Computational Fluid Dynamics Conference*, No. 4187, Miami, FL, June 2007.

[14]Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," *AIAA Paper 2008-0725*, Reno, NV, January 2008.

[15]Nemec, M. and Aftosmis, M. J., "Adjoint Sensitivity Computations for an Embedded-Boundary Cartesian Mesh Method," *J. Comp. Phys.*, Vol. 227, 2008, pp. 2724–2742.

[16]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[17]Anderson, G. R., Aftosmis, M. J., and Nemec, M., "Parametric Deformation of Discrete Geometry for Aerodynamic Shape Design," *AIAA Paper 2012-0965*, Nashville, TN, January 2012.

[18]Duvigneau, R., "Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization," Tech. Rep. 5949, INRIA, 2006.

[19]Nemec, M. and Aftosmis, M. J., "Toward Automatic Verification of Goal-Oriented Flow Simulations," Tech. Rep. TM-2014-218386, NASA, 2014.

[20]Morris, A. M., Allen, C. B., and Rendall, T. C. S., "Domain-Element Method for Aerodynamic Shape Optimization Applied to a Modern Transport Wing," *AIAA Journal*, Vol. 47, No. 7, 2009.

[21]Jakobsson, S. and Amoignon, O., "Mesh Deformation using Radial Basis Functions for Gradient-based Aerodynamic Shape Optimization," *Computers and Fluids*, Vol. 36, No. 6, July 2007, pp. 1119–1136.

[22]Rendall, T. C. S. and Allen, C. B., "Unified Fluid-Structure Interpolation and Mesh Motion using Radial Basis Functions," *Int. J. Numer. Meth. Eng.*, Vol. 74, 2008, pp. 1519–1559.

[23]Yamazaki, W., Mouton, S., and Carrier, G., "Geometry Parameterization and Computational Mesh Deformation by Physics-Based Direct Manipulation Approaches," *AIAA Journal*, Vol. 48, No. 8, August 2010, pp. 1817–1832.

[24]Wintzer, M., "Span Efficiency Prediction Using Adjoint-Driven Mesh Refinement," *Journal of Aircraft*, Vol. 47, No. 4, July-August 2010, pp. 1468–1470.

[25]Berger, M. and Aftosmis, M. J., "Progress Towards a Cartesian Cut-Cell Method for Viscous Compressible Flow," *AIAA Paper 2012-1301*, Nashville, TN, January 2012.

[26]Smith, S. C., Nemec, M., and Krist, S. E., "Integrated Nacelle-Wing Shape Optimization for an Ultra- High Bypass Fanjet Installation on a Single-Aisle Transport Configuration," *AIAA Paper 2013-0543*, Grapevine, Texas, January 2013.

[27]Campbell, R. L., "Efficient Viscous Design of Realistic Aircraft Configurations," *AIAA Paper 98-2539*, June 1998.

Appendix A. Optimization Details

Table 4 shows some of the optimizer and flow settings used for each case.

Table 4: Optimization and Flow Solver Parameters

	Case 1	Case 2	Case 3	Case 4
Farfield distance (x,y,z)	$(\pm 96, \emptyset, +96)$	$(\pm 96, \emptyset, \pm 96)$	$(\pm 48, +48, \pm 48)$	$(\pm 113, +113, \pm 113)$
Limiter	None	None	van Leer	van Leer
Trigger	Slope < 0.2	Optimality	Optimality	Optimality
SNOPT Major Step Limit	1.0	1.0	1.0	1.0